

AI & ML for IoT Systems

Why AI + IoT Now



Billions of connected sensors



Cheap compute at the edge



Real-time decisions are expected



AI turns raw telemetry into value

IoT Stack Overview



Perception layer: sensors, actuators



Network layer: connectivity + routing



Application layer: analytics, automation, AI



Feedback/control loop



From Raw Signals to Usable Data



Sampling rate
& Nyquist

Quantization
and
compression

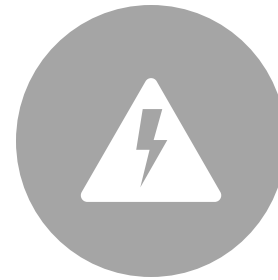
Denoising,
normalization

Time sync
across
devices

Control and Actuation Loops



Sense → infer → act



Closed-loop autonomy
(e.g. smart HVAC)



Latency budgets: ms vs
seconds



Safety and fail-safes



Feature Engineering for Sensor Data

- Windowing (fixed, sliding, adaptive)
 - Stats: mean, variance, kurtosis
 - Frequency domain: FFT, spectrograms
 - Domain features: vibration signatures
-

Why ML in IoT?

- Predictive maintenance
- Activity / gesture recognition
- Occupancy-aware energy optimization
- Anomaly detection in industrial systems
- Autonomous navigation / robotics

Edge vs Cloud vs Hybrid



Edge: instant response, local context



Cloud: heavy analytics, global context

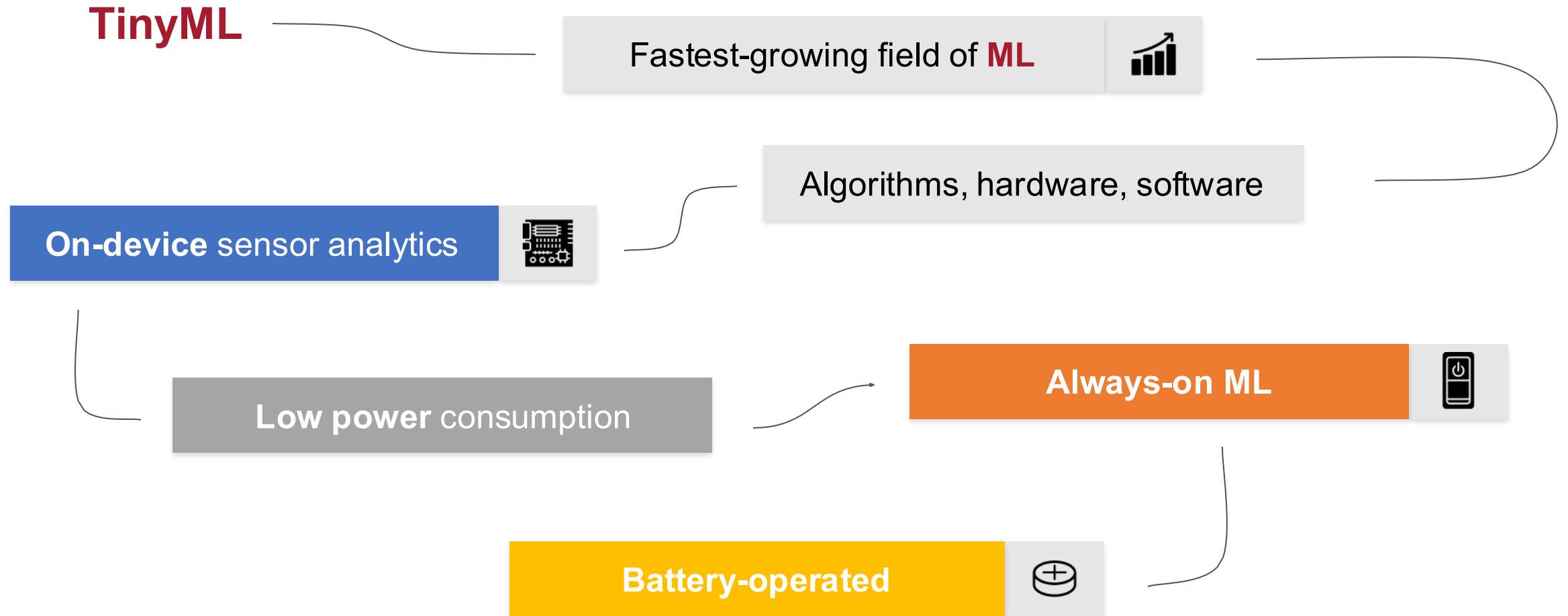


Hybrid: edge pre-filter + cloud refinement



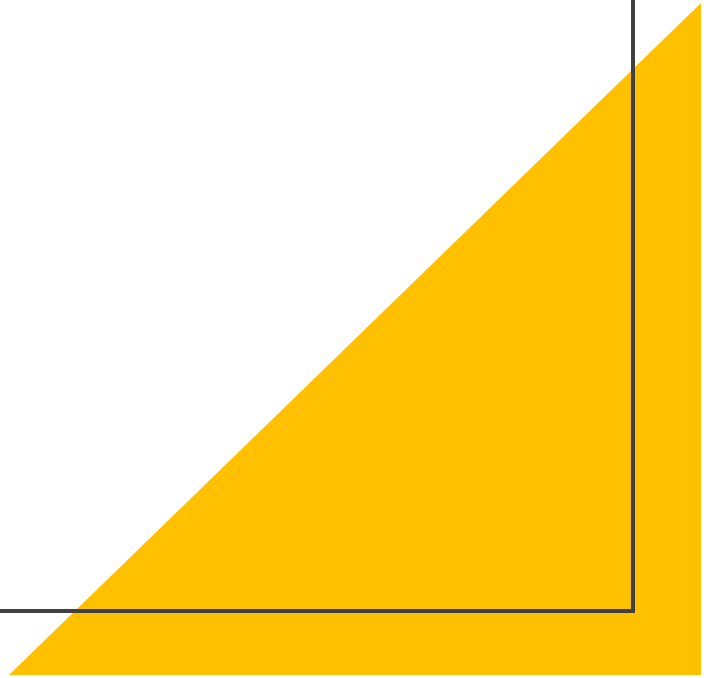
Cascade inference patterns

What is Tiny Machine Learning (**TinyML**)?

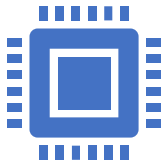


Why Edge Intelligence?

- Privacy: data stays local
- Low latency: sub-100 ms actuation
- Reliability offline / flaky network
- Bandwidth savings: send insights, not raw video



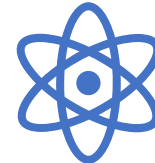
Edge Hardware Landscape



MCUs (tens of kB RAM)



Single-board
computers (Raspberry
Pi class)

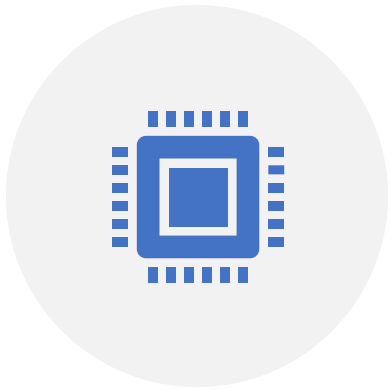


NPUs / TPUs /
accelerators



Battery and thermal
limits

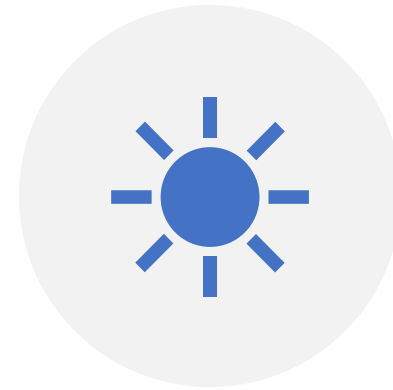
TinyML Concepts



ML ON ULTRA-LOW-POWER
MICROCONTROLLERS

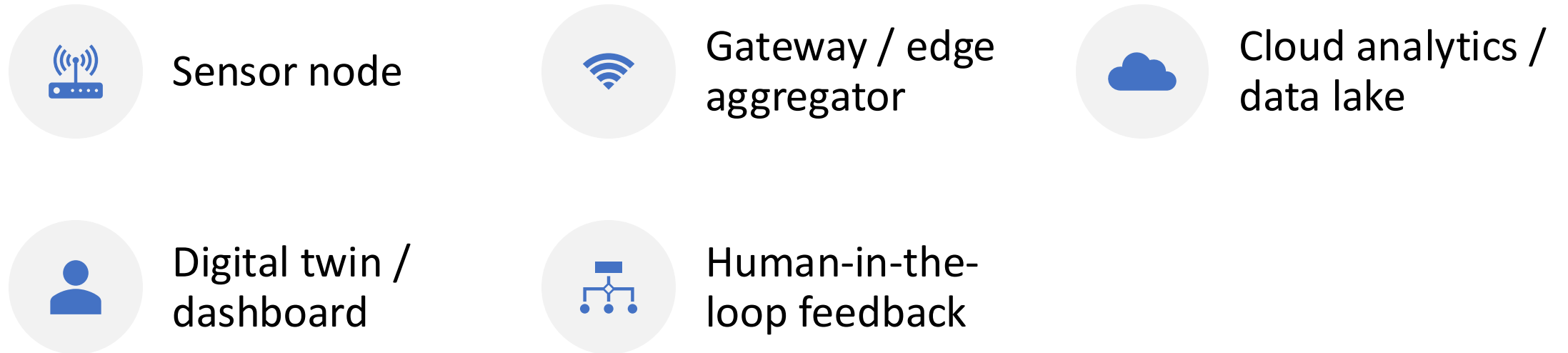


USE CASES: WAKE-WORD,
GESTURE, LEAK DETECTION



ALWAYS-ON SENSING
UNDER ~1 MW BUDGET

Reference AIoT Architecture



EdgeML (P↑)

Autonomous Car Control



Image Recognition



TinyML (P↓)

KeyWord Spotting



Environmental Control



Image Spot



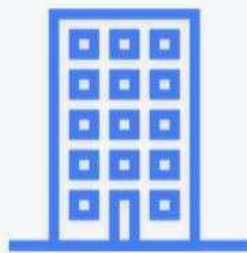
Motion & biometric



TinyML Application Areas



Home



Office



Industry



Predictive Maintenance



Motion, current, audio and camera

- Industrial
- White goods
- Infrastructure
- Automotive

Asset Tracking & Monitoring



Motion, temp, humidity, position, audio and camera

- Logistics
- Infrastructure
- Buildings

Human & Animal Sensing



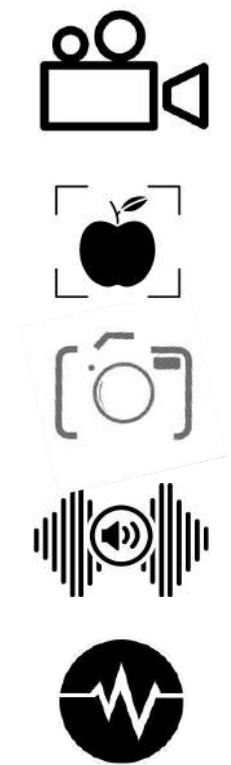
Motion, radar, audio, PPG, ECG

- Health
- Consumer
- Industrial

Hardware

EdgeML

TinyML



Anomaly Detection
Sensor Classification
20 KB

KeyWord Spotting
Audio Classification
50 KB

Image
Classification
250 KB+

Object Detection
Complex Voice
Processing
1 MB+

Video
Classification
2 MB+



Rpi-Pico
(Cortex-M0+)



Arduino Nano
(Cortex-M4)



Arduino Pro
(Cortex-M7)



RaspberryPi
(Cortex-A)

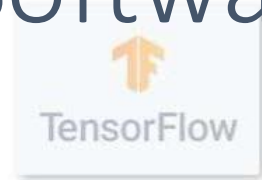


SmartPhone
(Cortex-A)



Jetson Nano
(Cortex-A + GPU)

Software



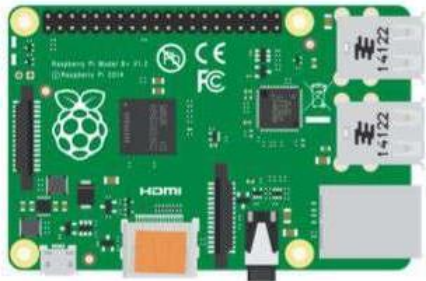
Train a model

Convert
model

Optimize
model

Deploy
model at
Edge

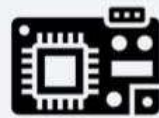
Make
inferences
at Edge



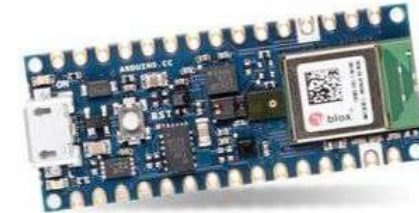
Raspberry Pi



Linux

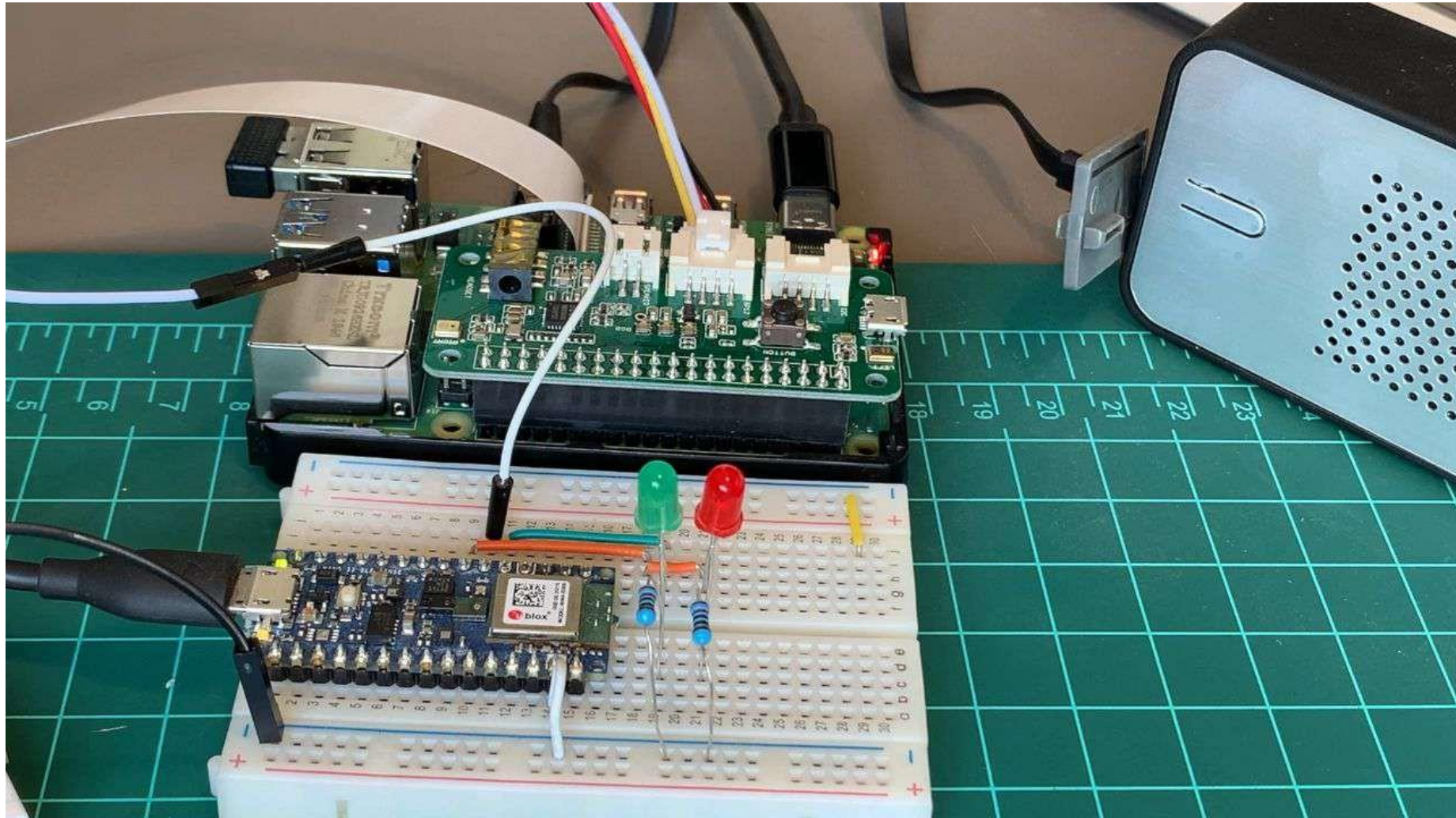


(TFL Micro)



Microcontroller

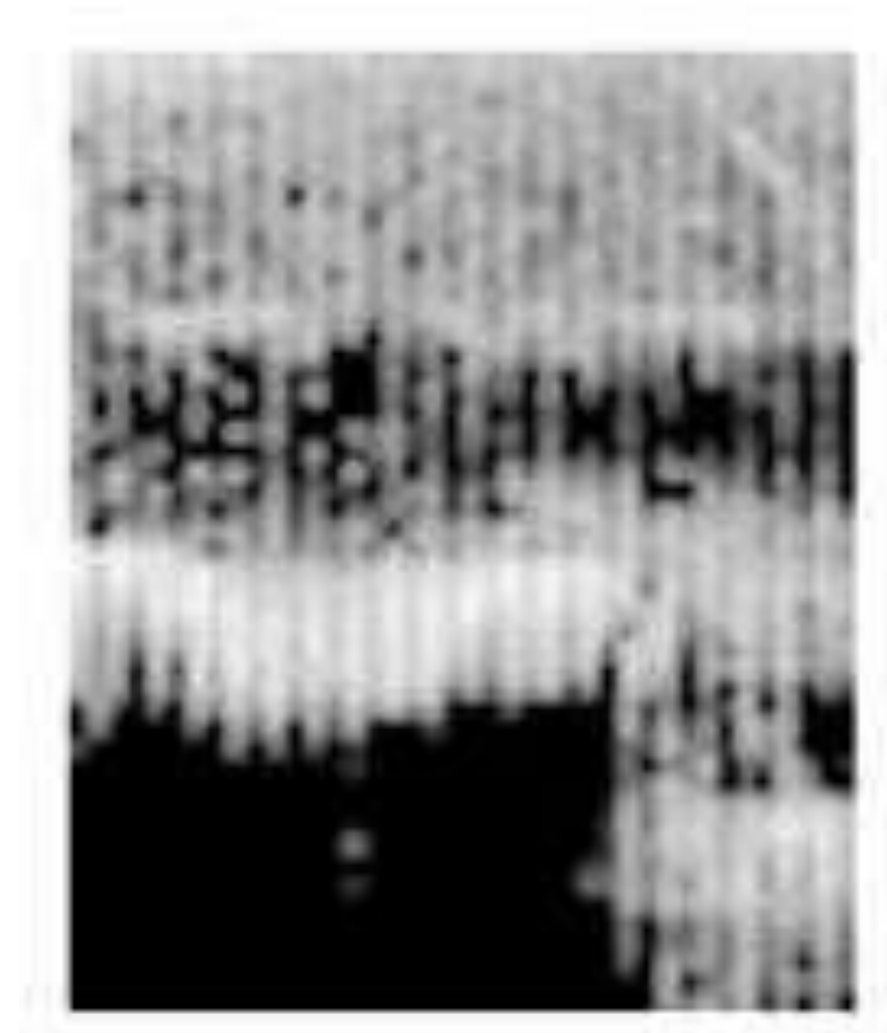
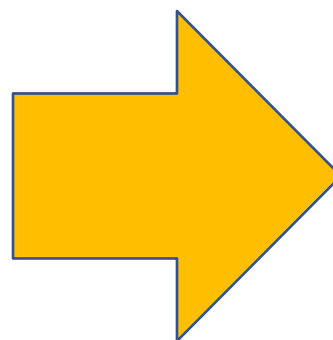
Example: KeyWord Spotting (KWS)



<https://mrobot.org/2021/01/27/building-an-intelligent-voice-assistant-from-scratch/>



Sound



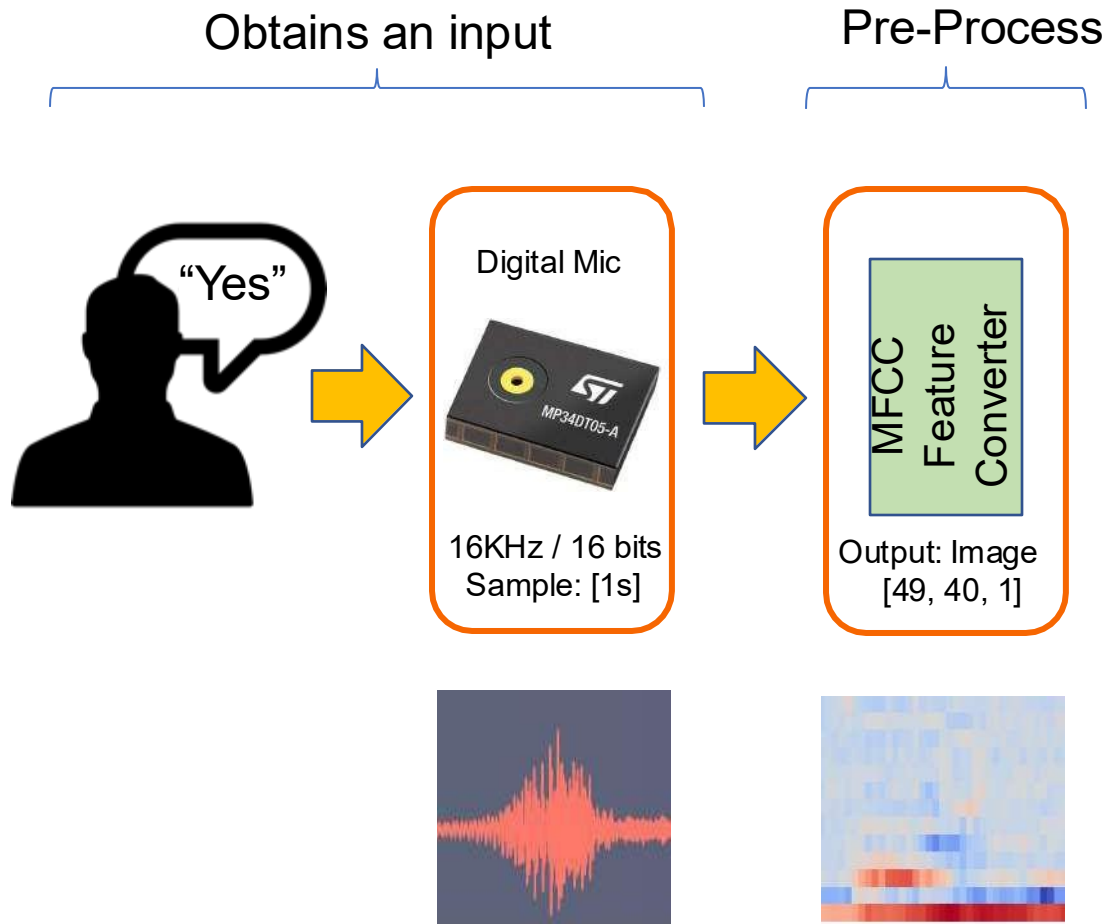
Image

KeyWord Spotting (KWS) - **Inference**

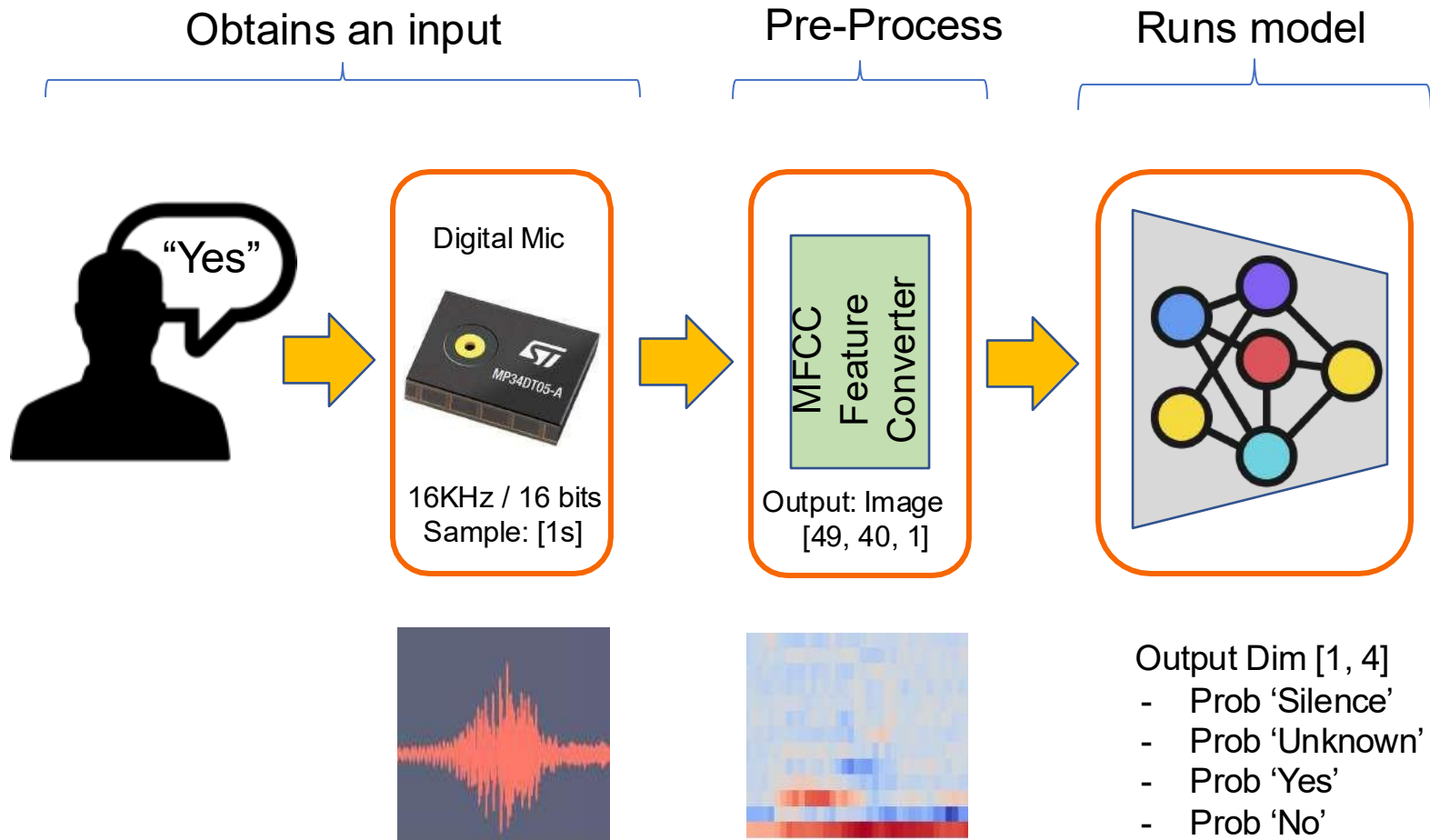
Obtains an input



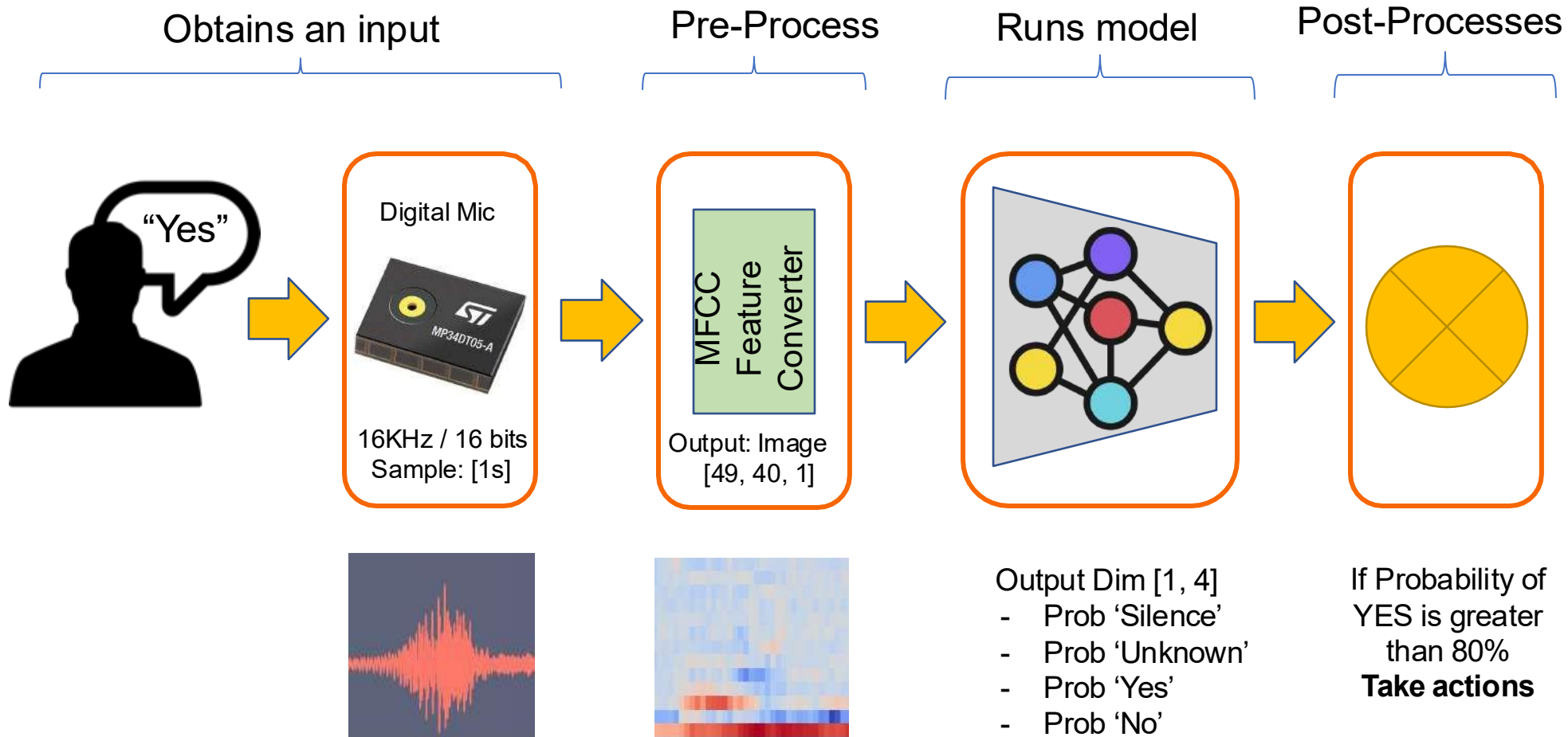
KeyWord Spotting (KWS) - **Inference**



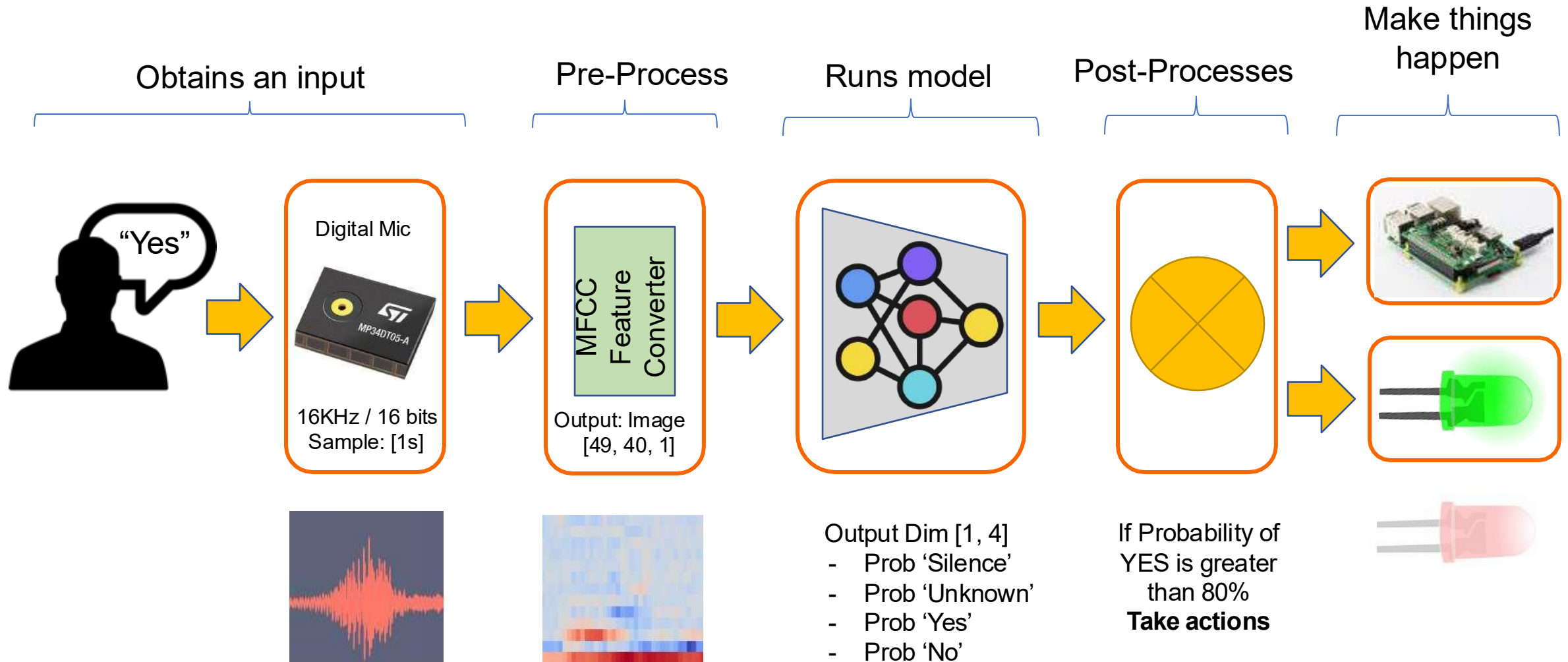
KeyWord Spotting (KWS) - **Inference**



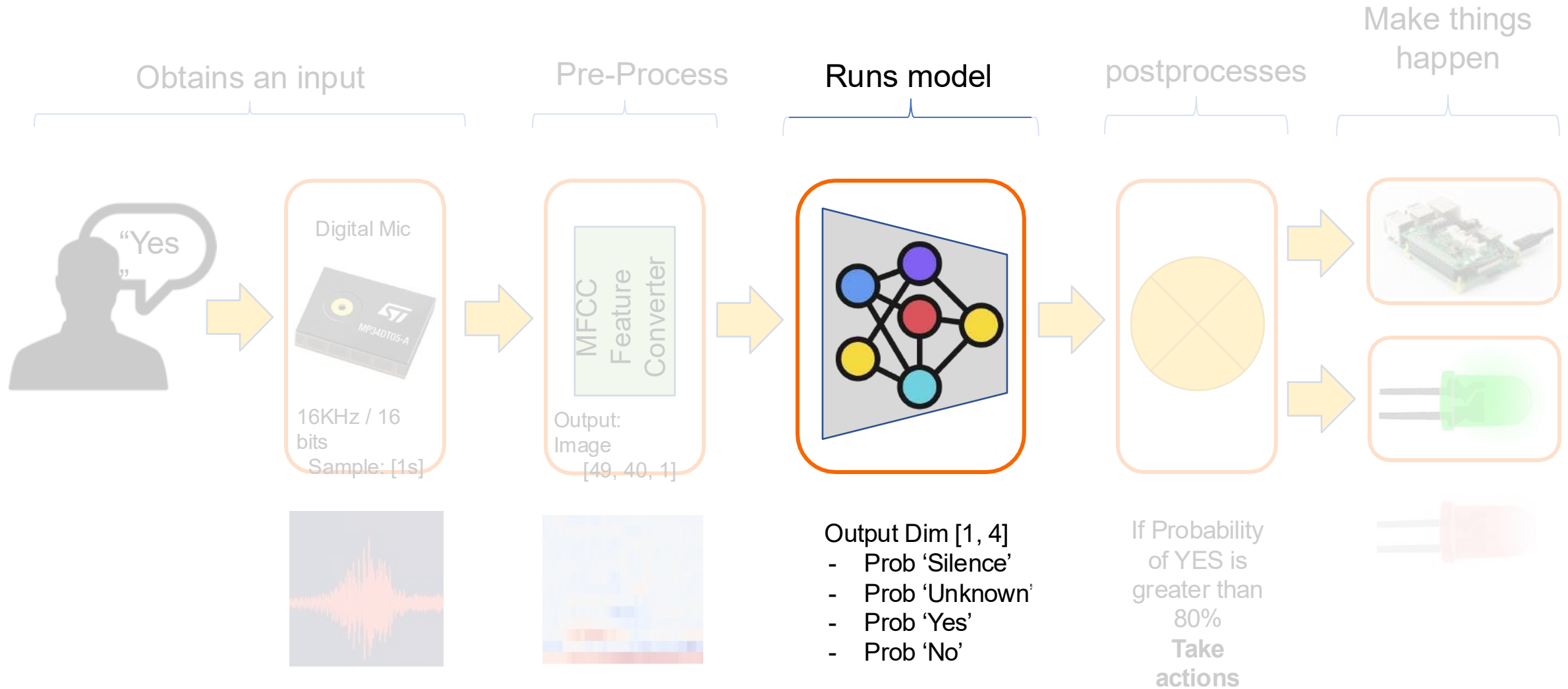
KeyWord Spotting (KWS) - **Inference**



KeyWord Spotting (KWS) - **Inference**



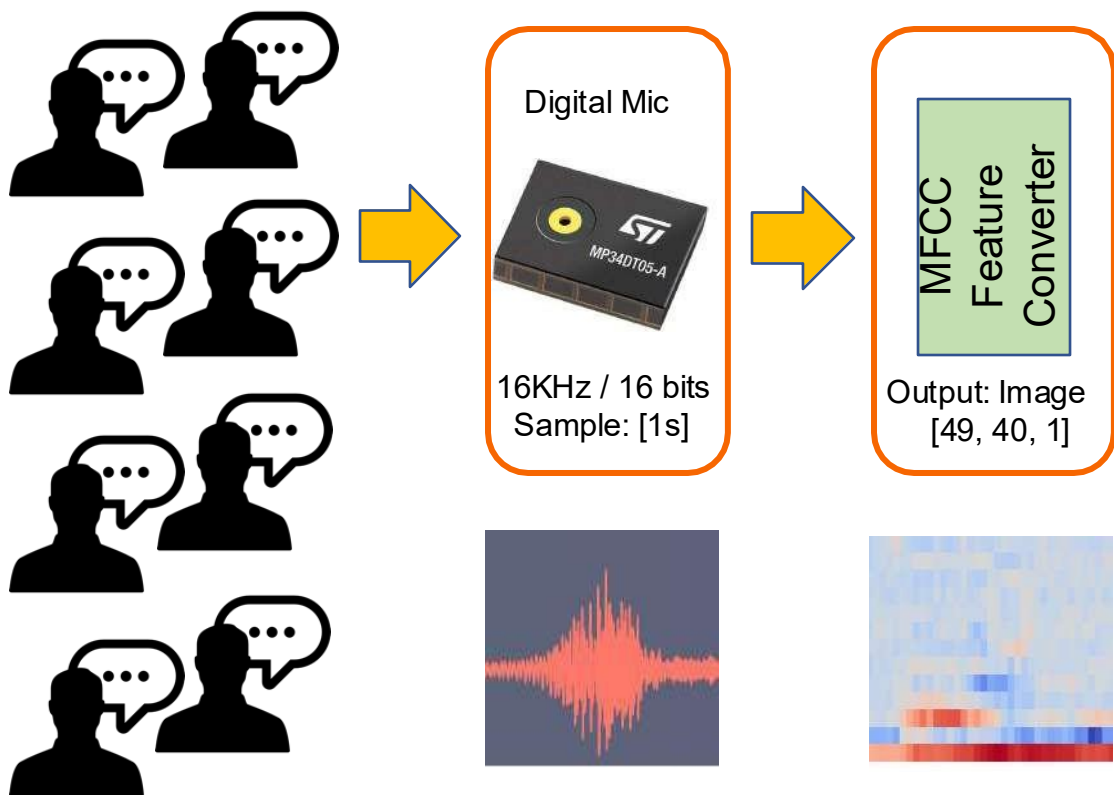
KeyWord Spotting (KWS) - **Model**



KeyWord Spotting (KWS) – **Create Model (Training)**

Obtains data

Pre-Process



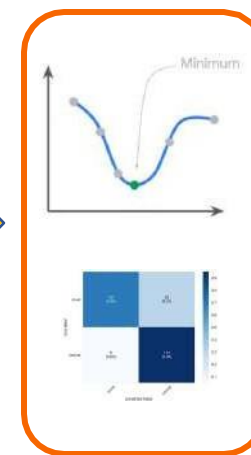
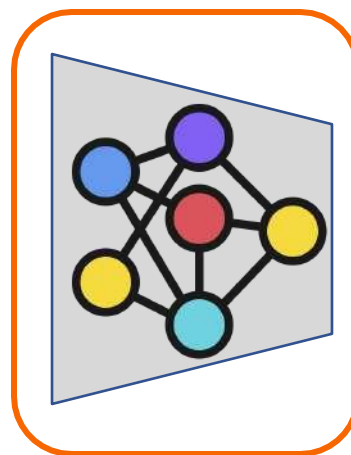
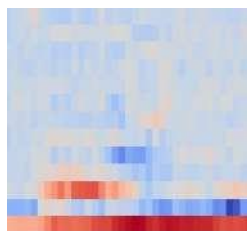
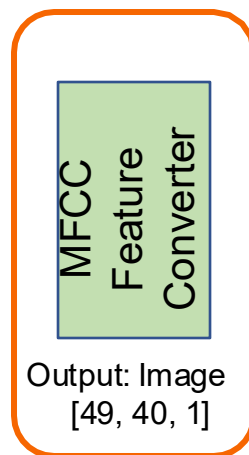
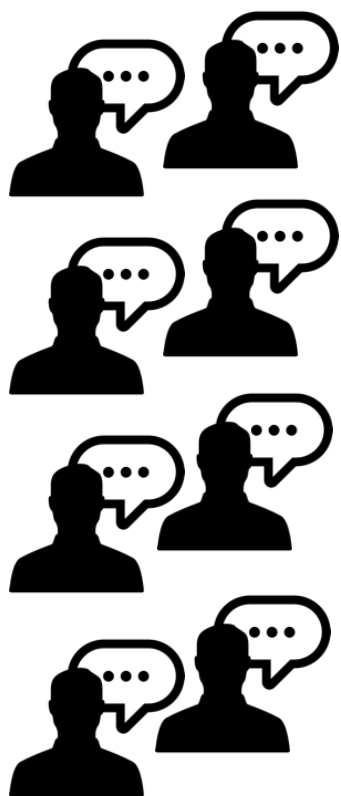
KeyWord Spotting (KWS) – **Create Model (Training)**

Obtains data

Pre-Process

Train model

Evaluate Model



KeyWord Spotting (KWS) – **Create Model (Training)**

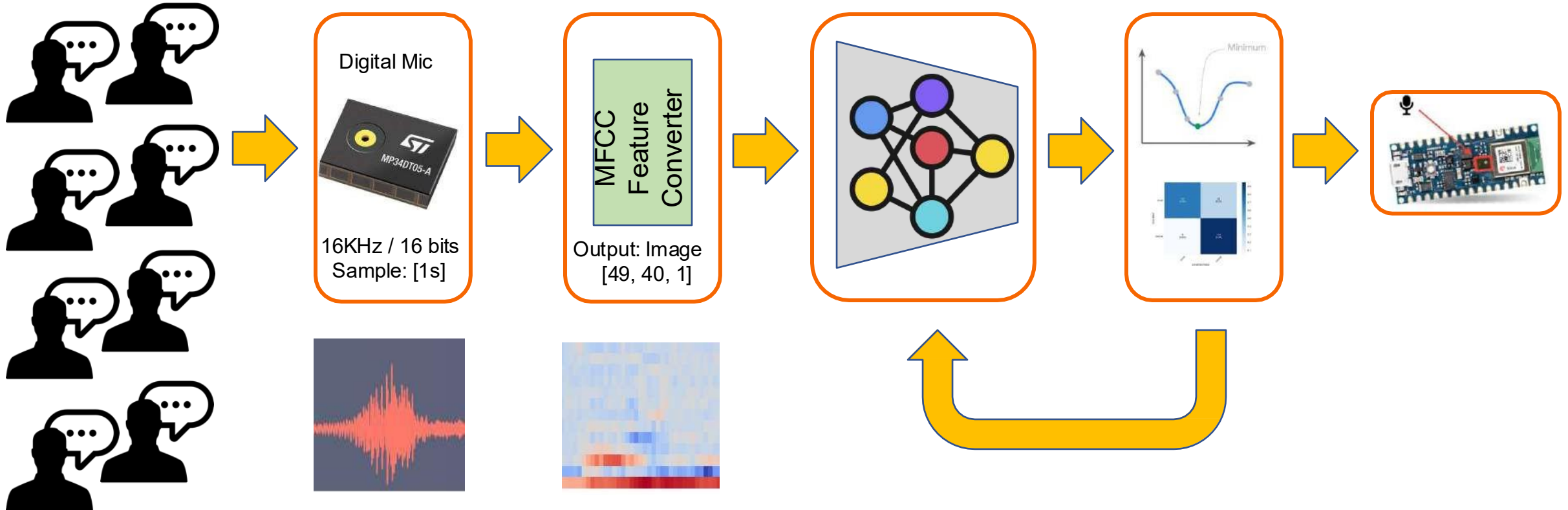
Obtains data

Pre-Process

Train model

Evaluate Model

Deploy



Supervised Learning Refresher

- Regression vs classification
- Temporal train/val/test splits matter
- Labeling is expensive and noisy
- Concept drift over time

Supervised Learning

- Majority of practical ML uses supervised learning
- Mapping function approximated from past experience
 - Regression $f(x)=y$, y is a real number
 - Classification $f(x)=y$, y is a category label
- Training
 - Labeled positive and negative examples
 - From unseen input predict corresponding output
 - Learning until acceptable performance is achieved

Unsupervised & Self-Supervised IoT



Clustering for behavior patterns



Autoencoders for anomaly detection



Contrastive learning on time windows



Few/zero-label environments

Unsupervised Learning

- Discover hidden relations and learn about the data
 - Clustering $f(X) = [X_1, \dots, X_k]$, k disjoint subsets
 - Association $f(X_i, X_j) = R$, relation
- Training
 - All examples are positive
 - No labeling, no teacher
 - No single correct answer
- Practical usage
 - Derive groups, not explicitly labeled
 - Market basket analysis (association among items)

Time Series Modeling Approaches



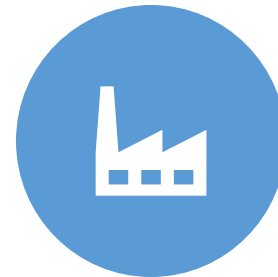
Classical: ARIMA,
Kalman filters



ML: Random Forests on
engineered windows



Deep: RNN / LSTM /
GRU / 1D CNN



Transformers for long
context

Activity Recognition Example

- Input: accelerometer / gyroscope
- Sliding window segmentation
- Feature extraction vs end-to-end deep model
- Output: walk / run / fall / idle
- Use case: elder fall detection

Predictive Maintenance Example

- Vibration + temperature from motors
- Estimate Remaining Useful Life (RUL)
- Early fault signatures are subtle
- Class imbalance: failures are $\sim 0.1\%$
- Metric: precision at very low false alarms

Anomaly Detection in IoT

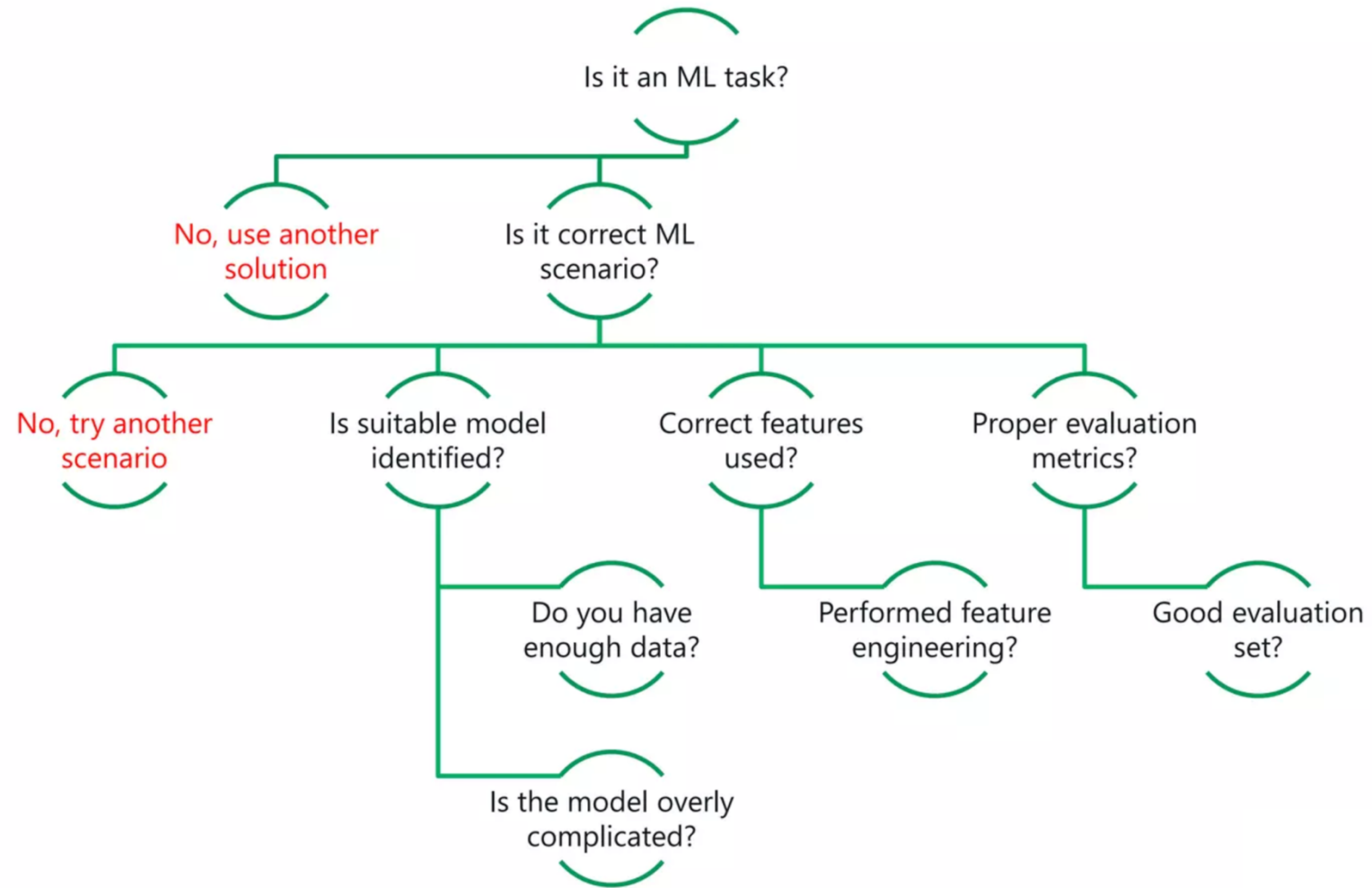
- Why anomalies matter: safety, cost, security
- Thresholding simple stats
- Isolation Forest / density methods
- Autoencoder reconstruction error
- Online streaming anomaly scores

Computer Vision in IoT

- Smart cameras: traffic, quality inspection
- Object detection, pose estimation
- PPE compliance / safety zones
- Privacy: blur faces, process on-device

Multimodal Sensor Fusion

- Fusing IMU + camera + GPS + audio
- Early fusion vs late fusion
- Attention-based fusion architectures
- Robustness if one sensor drops



ML Decision Tree

Diagnose Steps (part 1)

1. Is it a ML task? Are you sure ML is the best solution?

- Hard: X is independent of Y : $X = \langle \text{name, age, income} \rangle$, $Y = \text{height}$
- Easy: X is a set with limited variations. Configure $Y = F(X)$

2. Appropriate ML scenario?

- Supervised learning (classification, regression, anomaly detection)
- Unsupervised learning (clustering, pattern learning)

3. Appropriate model?

- Data size (small data \rightarrow linear model, large data \rightarrow consider nonlinear)
- Sparse data (require normalization to perform better)
- Imbalanced data (special treatment of the minority class required)
- Data quality (noise and missing values require loss function)

4. Enough training data?

- Investigate how precision improves with more data

Diagnose Steps (part 2)

5. Model overly complicated?

- Start simple first, increase complexity and evaluate performance
- Avoid overfitting to training set

6. Feature quality

- Have you identified all useful features?
- Use domain knowledge of an expert to start
- Include any feature that could be found and investigate model performance

7. Feature engineering

- The best strategy to improve performance and reveal important input
- Encode features, normalize [0:1], combine features

8. Combine models

- If multiple models have similar performance there is a chance of improvement
- Use one model for one subset of data and another model for the other

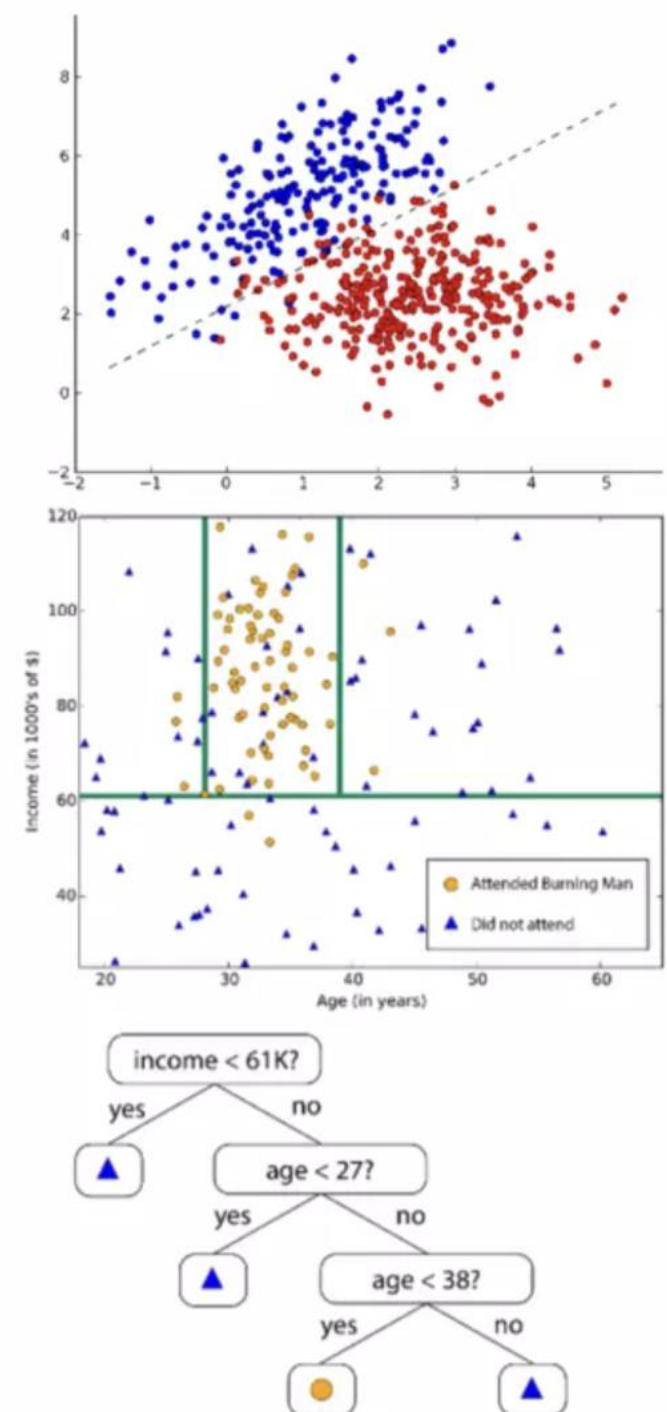
Diagnose Steps (part 3)

9. Model validation

- Use appropriate performance indicator (accuracy, precision, recall, F1, etc.)
- How well does the model describe data? (AUC)
- Data typically divide into Training and Validation
- Evaluate accuracy on disjoint dataset (other than training dataset)
- Tune model hyper parameters (i.e. number of iterations)

Types of Algorithms

- Linear Algorithms
 - **Classification** – classes separated by straight line
 - **Support Vector Machine** – wide gap instead of line
 - **Regression** – linear relation between variables and label
- Non-Linear Algorithms
 - **Decision Trees** and **Jungles** – divide space into regions
 - **Neural Networks** – complex and irregular boundaries
- Special Algorithms
 - **Ordinal Regression** – ranked values (i.e. race)
 - **Poisson** – discrete distribution (i.e. count of events)
 - **Bayesian** – normal distribution of errors (bell curve)



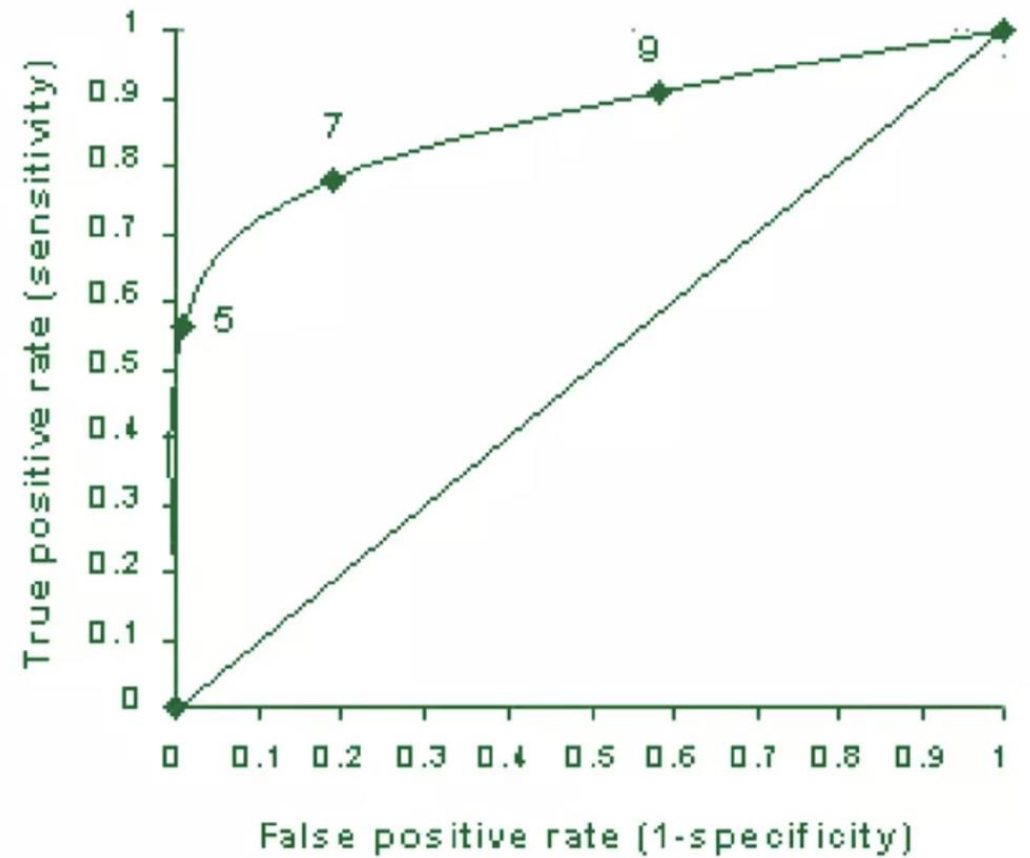
Model Performance (Classification)

- Binary classification outcomes {positive; negative}
- ROC curve
 - TP Rate = True Positives / All Positives
 - FP Rate = False Positives / All Negatives

- Example:

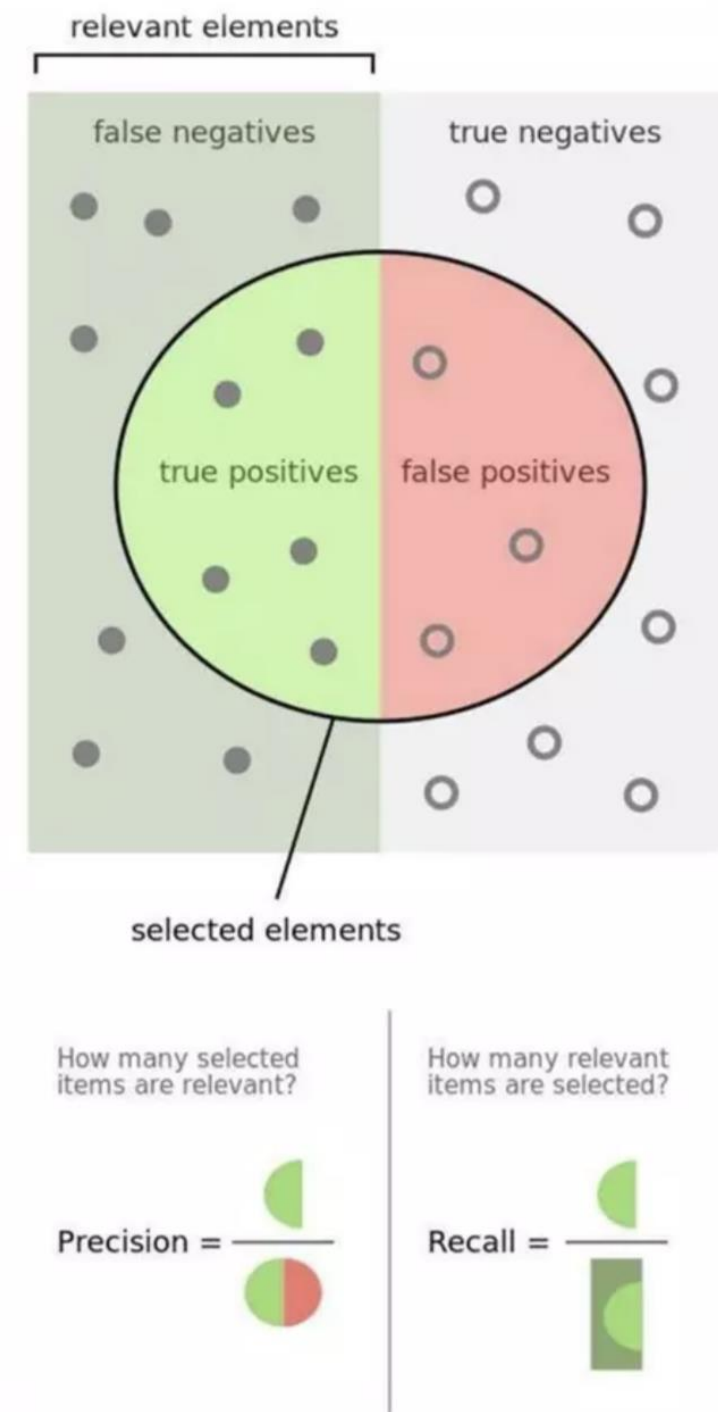
| | TP Rate | FP Rate | 1-FP Rate |
|---|---------|---------|-----------|
| 5 | 0.56 | 0.99 | 0.01 |
| 7 | 0.78 | 0.81 | 0.19 |
| 9 | 0.91 | 0.42 | 0.58 |

- AUC (Area Under Curve)
 - KPI for model performance and comparison
 - 0.5 = random prediction, 1 = perfect match
- For Multiclass – average from all RoC curves



Threshold Selection (Binary)

- Probability Threshold
 - Cost of one error could be much higher than cost of other
 - E.g. spam filter – it is more expensive to miss a real mail
- Accuracy
 - For symmetric 50/50 data
- Precision
 - E.g. 1000 devices, 5 fails, 8 predicted, 4 true failures
 - Correct positives (e.g. $4/8 = 0.5$, FP are expensive)
- Recall
 - Correctly predicted positives (e.g. $4/5=0.8$, FN are expensive)
- F1(balanced error cost)
 - Balanced cost of Precision/Recall



Model Performance (Regression)

- Coefficient of Determination (R^2)
 - Single numeric KPI – how well data fits model
 - $R^2 > 0.6$ – good, $R^2 > 0.8$ – very good, $R^2 = 1$ – perfect
- Mean Absolute Error (MAE) / Root Mean Squared Error
 - Deviation of estimates from observed real values
 - Compare model errors measure in the SAME units
- Relative Absolute Error Relative Squared Error
 - % deviation from real value
 - Compare model errors measure in the DIFFERENT units

Evaluation Metrics for IoT ML

- Latency (ms)
- Power draw (mW)
- Model size / memory footprint
- False alarm vs miss cost
- Closed-loop control stability

Model Compression Techniques

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Quantization (8-bit, 4-bit)
- Pruning weights / neurons
- Knowledge distillation
- Neural architecture search for low FLOPs

On-Device Inference Pipeline

- Capture → preprocess → inference → decision
- Memory budgeting for buffers + weights
- Real-time scheduling with firmware tasks
- Interrupt-driven ML (e.g. wake word)

Federated Learning for IoT

- Devices train locally on private data
- Only model updates are shared
- Benefits: personalization + privacy
- Challenges: non-IID data, stragglers

Digital Twins


A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Virtual replica of a physical asset
- Live sync with sensor data
- Simulate 'what if' scenarios
- Uses: predictive maintenance, optimization

Privacy & Responsible AI in IoT

- Always-on sensing = surveillance risk
- Behavior inference from 'harmless' signals
- Data minimization and edge filtering
- Transparency + consent for end users

Regulatory & Compliance

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Safety standards for industrial monitoring
- Data protection (GDPR-style thinking)
- AI accountability + audit trails
- Explainability for automated decisions

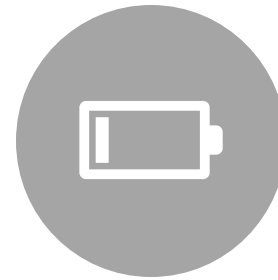
LLMs + IoT

- Natural language interfaces ('Why did the alarm trigger?')
- Edge copilots for technicians
- Summarizing fleets of sensor logs
- Risk: hallucination in safety-critical loops

Energy-Aware Intelligence



Adaptive sampling /
duty cycling



Battery-aware model
selection



Green AI: minimize
inference carbon cost



Sustainability as design
constraint

Edge Swarms & Collective Intelligence



Many small devices
collaborating locally



Gossip instead of
central cloud



Distributed anomaly
detection



Use cases: env
monitoring, precision
agriculture

Where This Field Is Going



Every sensor becomes intelligent



Every decision must be explainable



AI/ML is the differentiator in IoT



Your job: design systems that are smart, safe, and trustworthy